

Список использованных источников

1. Ландэ Д.В., Снарский А.А., Безсуднов И.В. Интернетика: Навигация в сложных сетях: модели и алгоритмы / Д.В. Ландэ. – М.: Книжный дом "ЛИБРОКОМ", 2009. – 264 с.
2. О методе оценивания промежуточных узлов передачи данных для маршрутизации в иерархических сетях разной топологии / А.В. Грачев, Т.В. Киселева, А.С. Добрынин, Р.С. Койнов // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. 2015. № 1. – С. 32–38.
3. О способе управления распределенной сетевой структурой и оценки её работы / А.В. Грачев, Т.В. Киселева // Труды Всероссийской научно-практической конференции с международным участием: Моделирование и наукоемкие информационные технологии в технических и социально-экономических системах. 2016. Ч. 2. – С. 188–192.

УДК 004.91

И. А. Гурин, Н. А. Спирин, В. В. Лавров, Н. И. Декун

ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина», г. Екатеринбург, Россия

МЕТОДЫ РАБОТЫ С ДОКУМЕНТАМИ MICROSOFT WORD ПРИ РАЗРАБОТКЕ ВЕБ-СЕРВИСОВ

Аннотация

Определены основные задачи работы с текстовыми документами Microsoft Word при разработке веб-сервисов. Рассмотрены основные методы создания и редактирования текстовых документов на языке программирования PHP. Представлена реализация и описание решения, позволяющего генерировать персонализированные документы на основе шаблона.

Ключевые слова: веб-сервис, язык программирования, PHP, автоматизация, текстовый документ, Microsoft Word, редактирование, генерация, PHPWord.

Abstract

The main tasks of working with Microsoft Word text documents when developing web services are determined. The main methods of creating and editing text documents in the PHP programming language are considered. The implementation and description of the solution allowing to generate personalized documents based on the template is presented.

Keywords: Web service, programming language, PHP, automation, text document, Microsoft Word, editing, generating, PHPWord.

Тенденция развития современных информационных технологий характеризуются переходом к облачным вычислениям и автоматизированной обработке данных. Если рассмотреть данный процесс с точки зрения работы с электронными документами, то он предполагает переход от работы с офисными приложениями на локальной машине пользователя к работе с облачными веб-сервисами, в которых происходит заполнение, обработка и генерация готовых текстовых документов, электронных таблиц и др. Документы, которые раньше создавались, редактировались и хранились локально, сейчас сохраняются на удаленных сетевых дисках.

Рассмотрим круг решаемых задач при автоматизированной обработке данных электронных документов.

Первая задача связана с персонализацией электронного документа, т. е. адаптацией электронного документа к конкретному пользователю. Персонализированный электронный

документ формируется путем заполнения заданного макета индивидуальной пользовательской информацией, представляет собой конечный информационный продукт и не предполагает внесение новой индивидуальной информации в документ. При решении данной задачи процедура взаимодействия сервера с пользователем является однонаправленной. Примерами таких задач является формирование следующих видов документов: счет, накладная, договор, резюме.

Вторая задача связана с многократным использованием промежуточного документа, который модифицируется пользователем и передается обратно на сервер. Такая задача является двунаправленной и предназначена для получения дополнительной информации от пользователя и ее последующего сохранения на сервере. Например, при создании анкет формируется документ, передаваемый пользователю. В предусмотренные в документе поля пользователь вносит свои данные и передает документ обратно на сервер. Полученные данные сохраняются на сервере вместе с имеющейся информацией данного пользователя и могут быть использованы при генерации других документов [1].

Первая и вторая задачи автоматизированной обработки данных предполагают работу с шаблоном электронного документа. Шаблон представляет собой форматированный текстовый документ, в котором участок изменяемых данных заменен на последовательность специальных символов, позволяющих однозначно определить данный участок.

При решении первой задачи выполняется запись данных в шаблон, а при решении второй задачи осуществляется и запись, и чтение данных.

Редактор текстовых документов Microsoft Word является одним из наиболее популярных текстовых процессоров на сегодняшний день. Поэтому в работе рассмотрены методы автоматизированной обработки текстовых документов Microsoft Word.

1. COM-объекты. COM (Component Object Model – Объектная модель компонентов) – технология от компании Microsoft, заключающаяся в создании программного обеспечения на основе различных взаимодействующих компонентов, каждый из которых доступен для использования многими программами и процессами. В современных версиях Windows COM используется очень широко. На основе COM были созданы технологии: Microsoft OLE Automation, ActiveX и другие [2]. Данная технология позволяет создавать и редактировать документы любой сложности, для работы используется язык программирования Visual Basic for Applications (VBA) и Microsoft Office API. Недостатками данной технологии являются:

- необходимость установки пакета Microsoft Office и, соответственно, необходимость использования операционной системы из семейства Windows;
- низкая скорость создания или открытия нового документа, т.к. работа осуществляется через запуск Microsoft Word в фоновом режиме;
- большая избыточность кода даже при создании простых документов.

2. Объекты .NET. Для работы с данными объектами на языке программирования PHP требуется установка программной платформы .NET Framework. Класс DOTNET позволяет создавать объекты из сборок .NET, вызывать их методы и использовать их свойства [3]. Данное решение повторяет недостатки первого метода, но значительно повышает читаемость кода, отвечающего за обработку документа за счет использования объектно-ориентированного подхода .NET. Другим преимуществом данного метода является переносимость кода, отвечающего за генерацию документа, между программами, написанными на различных языках программирования, но на единой платформе .NET.

3. Библиотека PHPWord. Библиотека позволяет создавать новые документы, редактировать шаблоны документов и имеет следующие возможности [4]:

- вставка и форматирование разделов документа;
- вставка и форматирование текста;
- вставка разрывов текста;
- вставка разрывов страниц;
- вставка и форматирование изображений и бинарных OLE-объектов;
- вставка и форматирование водяных знаков;

- вставка верхних и нижних колонтитулов;
- вставка и форматирование таблиц;
- вставка собственных названий и таблиц содержания;
- вставка и форматирование списка элементов;
- вставка и редактирование гиперссылок;
- работа с шаблонами.

Для работы библиотеки требуется PHP версии 5.2 или новее, расширения `xml` и `ZipArchive`. Ее недостатком является поддержка работы только с файлами формата DOCX.

При загрузке шаблона `PHPWord` позволяет заменять введенные в формате `{var1}` переменные на реальные строковые данные.

На рисунке 1 показаны шаблон документа и документ с данными, сформированный с использованием библиотеки `PHPWord`.

<p><u>PHPWord Template Example</u></p> <p>The following table contains a few values that can be edited by the <u>PHPWord Template</u> class. You just can edit single-line of text elements. The format and the rest of the document stay untouched.</p> <p>You just have to use the <u>PHPWord</u> search pattern like <u><code>\$(myReplacedValue)</code></u>.</p> <table> <tr><td rowspan="5">Data 1</td><td>Value 1:</td><td><code>\$(Value1)</code></td></tr> <tr><td>Value 2:</td><td><code>\$(Value2)</code></td></tr> <tr><td>Value 3:</td><td><code>\$(Value3)</code></td></tr> <tr><td>Value 4:</td><td><code>\$(Value4)</code></td></tr> <tr><td>Value 5:</td><td><code>\$(Value5)</code></td></tr> <tr><td rowspan="5">Data 2</td><td>Value 6:</td><td><code>\$(Value6)</code></td></tr> <tr><td>Value 7:</td><td><code>\$(Value7)</code></td></tr> <tr><td>Value 8:</td><td><code>\$(Value8)</code></td></tr> <tr><td>Value 9:</td><td><code>\$(Value9)</code></td></tr> <tr><td>Value 10:</td><td><code>\$(Value10)</code></td></tr> </table> <p>Today is <code>\$(weekday)</code> and it is <code>\$(time)</code>. Thanks for reading.</p>			Data 1	Value 1:	<code>\$(Value1)</code>	Value 2:	<code>\$(Value2)</code>	Value 3:	<code>\$(Value3)</code>	Value 4:	<code>\$(Value4)</code>	Value 5:	<code>\$(Value5)</code>	Data 2	Value 6:	<code>\$(Value6)</code>	Value 7:	<code>\$(Value7)</code>	Value 8:	<code>\$(Value8)</code>	Value 9:	<code>\$(Value9)</code>	Value 10:	<code>\$(Value10)</code>
Data 1	Value 1:	<code>\$(Value1)</code>																						
	Value 2:	<code>\$(Value2)</code>																						
	Value 3:	<code>\$(Value3)</code>																						
	Value 4:	<code>\$(Value4)</code>																						
	Value 5:	<code>\$(Value5)</code>																						
Data 2	Value 6:	<code>\$(Value6)</code>																						
	Value 7:	<code>\$(Value7)</code>																						
	Value 8:	<code>\$(Value8)</code>																						
	Value 9:	<code>\$(Value9)</code>																						
	Value 10:	<code>\$(Value10)</code>																						
<p><u>PHPWord Template Example</u></p> <p>The following table contains a few values that can be edited by the <u>PHPWord Template</u> class. You just can edit single-line of text elements. The format and the rest of the document stay untouched.</p> <p>You just have to use the <u>PHPWord</u> search pattern like <u><code>\$(myReplacedValue)</code></u>.</p> <table> <tr><td rowspan="5">Data 1</td><td>Value 1:</td><td>Sun</td></tr> <tr><td>Value 2:</td><td>Mercury</td></tr> <tr><td>Value 3:</td><td>Venus</td></tr> <tr><td>Value 4:</td><td>Earth</td></tr> <tr><td>Value 5:</td><td>Mars</td></tr> <tr><td rowspan="5">Data 2</td><td>Value 6:</td><td>Jupiter</td></tr> <tr><td>Value 7:</td><td>Saturn</td></tr> <tr><td>Value 8:</td><td>Uranus</td></tr> <tr><td>Value 9:</td><td>Neptun</td></tr> <tr><td>Value 10:</td><td>Pluto</td></tr> </table> <p>Today is Monday and it is 12:12. Thanks for reading.</p>			Data 1	Value 1:	Sun	Value 2:	Mercury	Value 3:	Venus	Value 4:	Earth	Value 5:	Mars	Data 2	Value 6:	Jupiter	Value 7:	Saturn	Value 8:	Uranus	Value 9:	Neptun	Value 10:	Pluto
Data 1	Value 1:	Sun																						
	Value 2:	Mercury																						
	Value 3:	Venus																						
	Value 4:	Earth																						
	Value 5:	Mars																						
Data 2	Value 6:	Jupiter																						
	Value 7:	Saturn																						
	Value 8:	Uranus																						
	Value 9:	Neptun																						
	Value 10:	Pluto																						

Рис. 1. Шаблон (слева) и документ с данными (справа), сформированный с использованием библиотеки `PHPWord`

Недостатком при работе с шаблонами в библиотеке `PHPWord` является, например, отсутствие возможности замены изображений. Поэтому предложен вариант, позволяющий как работать с переменными строками в шаблоне по аналогии с библиотекой `PHPWord`, так и осуществлять замену изображений.

4. Собственная реализация автоматизированной обработки текстовых документов. Документы Microsoft Word с расширением DOC являются бинарными файлами, а документы с расширением DOCX представляют собой файлы в открытом формате Office Open XML. Office Open XML – серия форматов файлов для хранения электронных документов пакетов офисных приложений – в частности, Microsoft Office. Формат представляет собой zip-архив, содержащий текст в виде XML, графику и другие данные [5].

Исходя из вышесказанного можно отметить, что работа с документами в формате DOC затруднена из-за отсутствия документации о структуре файла. Работа с файлами DOCX, наоборот, достаточно проста за счет открытости формата.

Единственным требованием для работы с документами с расширением DOCX, как и для работы библиотеки `PHPWord`, является наличие расширения `ZipArchive`.

Рассмотрим скрипт, выполняющий замену шести строковых переменных и одного изображения. Текст документа хранится в XML-файле `word/document.xml`, замена осуществляется строковой функцией `str_replace` без использования специализированных библиотек

для работы с XML. Замена изображения осуществляется удалением исходного и добавлением нового изображения в zip-архив под прежним именем. Изображения хранятся в папке *word/media*. Файл “my_photo.png” заменяет текущий файл “image1.png”. Далее приведен исходный код скрипта.

```
<?php

define('ROOTPATH', getcwd());

if( is_file(ROOTPATH . '/tmp/template_copy.docx') )
{
    unlink(ROOTPATH . '/tmp/template_copy.docx');
}
copy(ROOTPATH . '/template.docx', ROOTPATH . '/tmp/template_copy.docx');

$zip = new ZipArchive();

if( $zip->open(ROOTPATH . '/tmp/template_copy.docx') )
{
    $xmlstr = $zip->getFromName('word/document.xml');

    $xmlstr = str_replace('${FIO}', 'Гурин Иван Александрович', $xmlstr);
    $xmlstr = str_replace('${AGE}', 26, $xmlstr);
    $xmlstr = str_replace('${EDUCATION}', 'высшее', $xmlstr);
    $xmlstr = str_replace('${WORK_PLACE}', 'Уральский федеральный университет', $xmlstr);
    $xmlstr = str_replace('${POST}', 'аспирант', $xmlstr);
    $xmlstr = str_replace('${SKILLS}', 'Автоматизация, C#, PHP, MySQL', $xmlstr);

    $zip->addFromString('word/document.xml', $xmlstr);
    $zip->addFile(ROOTPATH . '/my_photo.png', 'word/media/image1.png');
    $zip->close();
}
?>
```

Результатом работы данного скрипта является создание нового файла с данными, который находится по адресу “/tmp/template_copy.docx”.

На рисунке 2 показаны шаблон документа и документ с данными, сформированный с использованием приведенного скрипта.



Рис. 2. Шаблон (слева) и документ с данными (справа), сформированный скриптом на языке программирования PHP

Представленный пример не является универсальным средством для работы с шаблонами документов Microsoft Word. Однако, он показывает простоту создания и обработки документов в формате Office Open XML. Этот формат, помимо текстовых документов, включает хранение электронных таблиц и презентаций.

Стоит отметить, что Office Open XML является форматом по умолчанию для приложений Microsoft Office 2007 и более поздних [5]. Однако, часто при документообороте встречаются документы с расширением DOC, созданные в пакете Microsoft Office 2003. Третий и четвертый методы решения не поддерживают работу с такими документами. В этом случае можно использовать конвертер, который первоначально преобразует файл с расширением DOC в файл формата Office Open XML с расширением DOCX. В качестве конвертера может использоваться пакет OpenOffice, обращение к которому производится из командной строки.

Список использованных источников

1. Миронов В.В. Информационная технология персонализации электронных документов Microsoft Office в WEB-среде на основе XML / В.В. Миронов, Г.Р. Шакирова, В.Э. Яфаев // Вестник Уфимского государственного авиационного технического университета. 2008. №2. С. 112-122.
2. Сухов К. Возможности языка PHP. Работа с приложениями посредством технологии COM / К. Сухов. Системный администратор. 2010. № 3. С. 70-74.
3. DOTNET Manual [Электронный ресурс] // The PHP Group: [web-сайт]. – Режим доступа: <http://php.net/dotnet> (дата обращения 12.04.2017).
4. PHPWord [Электронный ресурс] // CodePlex: хостинг проектов для открытого программного обеспечения: [web-сайт]. – Режим доступа: <https://phpword.codeplex.com/> (дата обращения 12.04.2017).
5. Office Open XML [Электронный ресурс] // Википедия. Свободная энциклопедия: [web-сайт]. – Режим доступа: https://ru.wikipedia.org/wiki/Office_Open_XML (дата обращения 12.04.2017).

УДК 004.91

С. А. Дианов, В. Г. Лисиенко

ФГАОУ ВО «Уральский федеральный университет имени первого Президента России Б.Н. Ельцина», г. Екатеринбург, Россия

ПОИСК ВЫБРОСОВ В МНОГОМЕРНОМ МАССИВЕ ДАННЫХ

Аннотация

В докладе изложена проблема наличия выбросов в базах данных производственных процессов. Рассмотрена процедура обработки данных, направленная на поиск выбросов. Произведено сравнение различных методов поиска выбросов.

Ключевые слова: нахождение выбросов, предварительная обработка данных, идентификация процесса, робастность.

Abstract

The problem of outliers in industrial processes databases is outlined. The outlier detection data preprocessing procedure is examined. The different ways of outlier detection are compared.

Keywords: outlier detection, data preprocessing, process identification, robustness.

Моделирование технологических процессов методом «чёрного ящика» требует сбора большого количества данных, полученных путём сохранения мгновенных значений факторов и откликов в базу данных. Такие методы моделирования подвержены значительному искажениям, связанным с содержащимися в выборке резко выделяющихся наблюдений. Они могут возникать из-за несовершенства измерительного оборудования, нестабильности процесса, человеческого фактора. Все эти обстоятельства ведут к затруднению идентификации производственного процесса. Поэтому перед моделированием эти данные